CRYPTO

Version 3.0

FIPS 140-2

Level 1 Validation

Security Policy

January 8, 2003

Table of Contents

1 INTRODUCTION

1.1 Purpose

This document is a Security policy for the Phaos Crypto version 3.0 Module and was produced as a part of the Federal Information Processing Standard (FIPS) 140-2 level 1 validation process.

This Security Policy document describes how version 3.0 of the Phaos Crypto Module meets all the requirements for level 1 validation criteria specified in FIPS PUB 140-2.

1.2 Organization

Section 1 of this document provides an overview of this Security Policy document. Section 2 describes the Phaos Crypto Module and how it meets the FIPS 140-2 requirements. Section 3 describes the usage of the Phaos Crypto Module to satisfy the approved FIPS 140-2 Level 1 mode of operation.

2 PHAOS CRYPTO PRODUCT

The Phaos cryptography products are unique in that they provide Java developers with a lightweight, extensible framework for integrating the full range of cryptography tools into any Java application or applet.

Phaos Crypto allows developers to protect data using FIPS Approved algorithms. It forms the core of the Phaos product line and provides cryptographic services to Phaos's high-level security protocol toolkits like Phaos Security Engine, Phaos SSLava, Phaos S/MIME, Phaos XML Security, Phaos SAML, Phaos Liberty, Phaos Centuris PKI, Phaos XKMS and Phaos Financial Engine.

2.1 Phaos Crypto Module

Phaos Crypto is a FIPS 140-2 Level 1 compliant, software based cryptographic module. It is a pure Java Toolkit that implements state of the art cryptographic algorithms, which are accessible through an easy-to-use API. Software developers can dynamically link the Phaos Crypto module into their applications to provide FIPS 140-2 compliant cryptographic support.

Phaos Crypto consists of a Java class library package as signed Java Archive (JAR) file. Phaos Crypto is classified as a multi-chip standalone module for FIPS 140-2. The actual cryptographic boundary includes the Crypto Module running in a Java Runtime Environment compatible with JDK 1.1 installed upon a PC running on the Windows 2000 Operating System. The physical interfaces of a general PC include the keyboard, mouse, monitor, serial ports, parallel ports, and network adaptors. The power interface for a general PC is external.

The Crypto Module is not capable of outputting any data beyond the physical cryptographic boundary. All data and services provided by the Crypto Module are output to the calling application that resides on the host machine. The Java Toolkit Module was validated as meeting all applicable FIPS 140-2 security requirements.

Phaos Crypto can run on any Java Runtime Environment compatible with JDK 1.1 (or higher) but was not tested upon each of these Java environments as a part of the validation

process.

Shown above is the software block diagram of the cryptographic module.

2.2 Module Interfaces

The logical interface to the Phaos Crypto Module is a Java Language Application Program Interface (API) documented in the Phaos Crypto API Docs.

Data Input and Data Output are provided with variables passed within the API function calls. The Status Output for each API function call is provided in the returned variables and exceptions that are documented for the API function call. The Status Output for the Phaos Crypto Module is provided through the ModuleState Class (com.phaos.fips package) API functions. The Control Input is provided thru the actual calls to the module functions.

The Phaos Crypto Module API is the logical interface for Data Input, Data Output and Status Output, and meets the Level 1 validation requirements of FIPS 140-2.

2.3 Roles, Services and Authentication

The Phaos Crypto Module implements a single role, operator, that combines the User and Crypto Officer roles. As allows by FIPS 140-2 Level 1, the Phaos Crypto Module does not support operator identification or authentication for the operator. An operator can performall services supported by the module.

The Phaos Crypto Module is intended to run on Windows 2000 in single user mode. When run in this configuration, no concurrent operators are supported.

Because the module is a Java class library, each process requesting access is provided its own instance of the class objects in the module. As a result, each process has complete access to all the information and keys within the class objects. However, no keys or other information are maintained when the class goes out of scope and is zeroized. Thus, an

instantiation of the class objects will only contain information placed in it by the process.

All the services offered are accessible to the operator, the only role supported by the Phaos Crypto Module. The services are broken into five packages, which are described below.

com.phaos.ASN1

Provides facilities for reading and writing both BER (Basic Encoding Rules) and DER (Distinguished Encoding Rules) encoded Abstract Syntax Notation One (ASN.1) structures. It supports all the ASN.1 datatypes defined in the ITU-T Recommendations X.208 and X.209. Also included is a streams based interface for one-pass reading and writing of ASN.1 structures.

com.phaos.crypto

Provides a full suite of cryptographic algorithms, including message digests such as MD5, SHA-1, SHA-256/384/512, symmetric encryption algorithms such as AES and Triple-DES, public key algorithms such as RSA, DSA and ECC, authentication algorithms such as HMAC, key agreement such as Diffie-Hellman and pseudo-random number generation. It also supports the public key cryptography standards (PKCS) #1, #5 and #8. It also provides facilities for symmetric key and asymmetric key-pair generation.

com.phaos.fips

Provides self-tests and utility classes for the FIPS 140-2 approved mode of operation. It allows for the explicitly invocation of the various power-up and conditional self-tests by the operator role. Also provided are facilities for

determing the operational state of the Phaos Crypto module.

com.phaos.math

Provides utility classes for mathematical functions. It provides for randomized primality testing and generation

com.phaos.utils

Provides utility classes for the other packages. It provides convenient access to some of the commonly used functionality. Some of the functionality provided includes Object pooling, Unsynchronized Streams and Collections, and Datatype conversions.

The Phaos Crypto Module does not support any authentication mechanisms.

2.4 Operational Environment

The operational environment of the Phaos Crypto Module is a general-purpose operational environment. The Crypto module uses an approved integrity verification technique as a part of the Power-Up Self Tests.

Phaos Crypto is classified as a multi-chip standalone module for FIPS 140-2. The actual cryptographic boundary includes the Crypto Module running in a JDK 1.1 compatible Java Runtime Environment installed upon a PC running on the Windows 2000 Operating System. The Java Toolkit Module was validated as meeting all applicable FIPS 140-2 security requirements.

Phaos Crypto can run on any JDK 1.1 (or higher) compatible Java Runtime Environment but was not tested upon each of these Java environments as a part of the validation process.

2.5 Cryptographic Key Management

2.5.1 Random Number Generators

The Phaos Crypto module contains 2 entropy harvesting and 3 pseudo-random number generator (PRNG) implementations. It includes an approved RNG algorithm described in section 3.1 of FIPS PUB 186-2 along with the modifications specified in the Random Number Generation and General Purpose Number Generation sections of Change Notice 1 for FIPS PUB 186-2. In order to use the module in the Approved mode, only the DSARandomBitsSource object should be used to generate random numbers.

The other two PRNG implementations (MD5RandomBitsSource and SHA1RandomBitsSource) should not be used as they are not Approved PRNGs.

The 3 PRNG implementations and 2 entropy harvesting implementations successfully passes the statistical random number generator tests described in section 4.9.1 of FIPS PUB 140-2 for quality assurance.

2.5.2 Key Generation

The Phaos Crypto module supports the generation of symmetric and asymmetric keys using the FIPS approved pseudo-random number generator. The module generates the following Approved asymmetric keys: RSA (PKCS#1), DSA, Diffie-Hellman (DH) and Elliptic Curve (EC) public and private keys.

The module is also capable of generating the following Approved symmetric keys: AES (128/192/256), DES, and Triple-DES.

No intermediate key generation values are output from the module during or upon completion of the key generation process.

2.5.4 Key Entry and Output

Phaos Crypto module allows for the import and export of keys through the use of API methods.

The Phaos Crypto Module requires that two internal boolean flags be set to true before plaintext cryptographic key components or other CSPs may be output.

2.5.5 Key Storage

The Phaos Crypto module does not support persistent storage of key material. The operator is responsible for the storage of keys in a manner that meets the FIPS 140-2 Level 1 security requirements.

A special purpose key is stored in the Phaos Crypto Module. Embedded in the module is a single DSA public key that is used to validate the integrity of the module’s bytecode.

2.5.6 Key Zeroization

The Phaos Crypto Module automatically zeros any keys in memory when the objects are no longer used and before garbage collection by the Java Runtime. The erase() API function call is provided, which the operator can invoke to explicitly zero key material.

2.5.7 Security Relevant Data Items

The following is a list of all security relevant data items (SRDI) used for both Approved and non-Approved security functions within the Phaos module:

SRDI	Description	Access
Symmetric keys	Symmetric keys used for encryption and decryption services	User and Crypto-officer role (read, write, and delete)
Asymmetric keys	Asymmetric keys used for signing and verifying messages	User and Crypto-officer role (read, write, and delete)
HMAC keys	Shared symmetric key used for computing HMAC	User and Crypto-officer role (read, write, and delete)

2.6 Self Tests

2.6.1 Power-Up Self Tests

When the Phaos Crypto Module is used for the first time, it executes self-tests to ensure integrity of the module and correct operation of the cryptographic algorithms. Self tests can also be run by the operator using the powerUpSelfTest() API function call in the com.phaos.fips package. When the self-tests are executing, the module suspends all processing until the self-tests complete. Failure of any of the power-up self-tests results in the module transitioning to an

unrecoverable error state. In the unrecoverable error state, the module is not operational and cannot perform any cryptographic services.

The following self tests are performed:

- DES known-answer test
- Triple-DES known-answer test
- AES known-answer test
- SHA-1 known-answer test
- HMAC-SHA1 known-answer test
- Random Number known-answer test
- DSA sign/verify test
- RSA encrypt/decrypt and sign/verify test
- ECDSA sign/verify test
- Module Integrity test using DSA signatures

2.6.2 Conditional Tests

Conditional tests are run when the conditions specified in FIPS 140-2 occur. Failure of any of the tests results in the module transitioning to a recoverable error state. Upon failure of a conditional self-test, the module enters the error state and throws an exception. The operator may optionally try and generate another random number/key pair or install a fresh version of the module.

The following self tests are performed:

- Pair wise consistency test for RSA, DSA, DH and EC key pairs
- Continuous random number generation test

2.7 Mitigation of Other Attacks

The Phaos Crypto module is designed to prevent generation of key material that is considered to be weak.

The Phaos Crypto module also supports specifying a high iteration count (default value of 1000) to thwart PKCS#5 password based encryption (PBE) dictionary attack, facilities for using RSA-OAEP to prevent Daniel Bleichenbacher's PKCS#1 attack as well as secure serialization of Java objects to ensure that keys and passwords are not serialized. However, these services are not allowed for use in the Approved mode of operation.

3 MODE OF OPERATION

3.1 Cryptographic Algorithms

The Phaos Crypto Module supports a wide range of cryptographic algorithms. FIPS 140-2 requires that only FIPS approved algorithms must be used in the approved mode of operation. The table below shows both the FIPS approved and non-approved algorithms provided in the Phaos Crypto Module.

	Algorithm	FIPS
Symmetric Ciphers	AES	FIPS 197
	DES	FIPS 46-3 (May only be used for legacy systems)
	Triple DES	FIPS 46-3
	RC2	No FIPS Standard
	RC4	No FIPS Standard
	Blowfish	No FIPS Standard
Message Digests	MD2	No FIPS Standard
	MD4	No FIPS Standard
	MD5	No FIPS Standard
	SHA-1	FIPS 180-1
	SHA 256/384/512	FIPS 180-2 (May not be used in the Approved mode of operation)
MAC	HMAC SHA-1	FIPS 198
Random Number Generation	MD5	No FIPS Standard
	SHA-1	No FIPS Standard
	DSA	FIPS 186-2
RSA	RSA Encryption /Decryption	No FIPS Standard
	RSA Signatures (PKCS#1)	FIPS 186-2
DSS	DSA Signatures	FIPS 186-2
Diffie-Hellman	DH Key Agreement	No FIPS Standard (May be used in Approved mode of operation)

Elliptic Curves	ECDH Key Agreement	No FIPS Standard (May be used in Approved mode of operation)
	ECDSA Signatures	FIPS 186-2

All FIPS approved algorithms have been validated as part of the FIPS 140-2 Level 1 validation process. Only FIPS Approved algorithms may be used in the Approved mode of operation.

3.2 FIPS 140-2 Level 1 Approved Mode of Operation

The Phaos Crypto module does not require a configuration to operate in a FIPS 140-2 approved mode. Section 3.1 lists the cryptographic algorithms provided by the Phaos Crypto module.

FIPS 140-2 requires that only FIPS approved algorithms be used while operating the module in a FIPS 140-2 mode. As a result, only the FIPS approved algorithms listed in section 3.1 must be used in the FIPS 140-2 approved mode of operation.

To use the Phaos Crypto module in a FIPS 140-2 Level 1 approved mode of operation, it is the responsibility of the operator to ensure the following:

- Use FIPS approved algorithms only
- Use the FIPS approved PRNG only (DSARandomBitsSource)
- DES algorithm should only be used for legacy systems
- RSA encrypt/decrypt may only be used for key transport
- Do not use deprecated API methods or fields
- Do not use the SHA-2 algorithm for hashing
- Do not use keys derived from passwords (PKCS#5 or PKCS#12 SafeBag) for key or data encryption
- Seed with material that is greater than the bit length of key during key generation
- Do not modify the OIDManager settings